

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 21 (2014) 27 – 33

www.elsevier.com/locate/procedia

24th CIRP Design Conference

Communication in multidisciplinary systems architecting

G. Maarten Bonnema^a

^aDepartment of Design, Production and Management, Faculty of Engineering Technology, University of Twente, Enschede, The Netherlands, P.O. Box 217, NL-7500 AE Enschede, The Netherlands

* Corresponding author. Tel.: +31-53-489-2548; fax: +31-53-489-3631. E-mail address: g.m.bonnema@utwente.nl

Abstract

Systems architecting is multidisciplinary by nature. It is interesting to note that the methods and tools that are developed and presented in literature are mostly based on one or a very limited number of formalisms. This means that an often large part of the stakeholders involved in the architecting process are hindered in the understanding of, and contributing to the architecture.

The paper investigates the architecting process and complexity in combination with knowledge and knowledge creation. Communication is identified as essential. It thus follows that tools that base on one formalism limit this communication in a multidisciplinary setting. Based on experiences from architects, literature and the author, ingredients for successful multidisciplinary architecting are listed, and directions for future research in complex systems architecting are given, based on these ingredients.

© 2014 Elsevier B.V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Selection and peer-review under responsibility of the International Scientific Committee of “24th CIRP Design Conference” in the person of the Conference Chairs Giovanni Moroni and Tullio Tolio

Keywords: Complex systems architecting; Research; Communication

1. Introduction

Systems architecting is getting increasingly complicated due to a variety of reasons. On the one hand the number and complexity of product functions are increasing. The application area of the systems under design is becoming more complex, too. On the other hand, we observe the increasing number of stakeholders as well as their disciplines. Architecting is predominantly a non-deterministic search process, where the outcome cannot fully be anticipated. Although the goal of the architecting process *can* be clear (it is often not), the system that it will produce is difficult to foresee from the outset.

As systems architecting is in essence *multidisciplinary*, it is performed by people with diverse backgrounds. Partial solutions in different domains have to be weighed and balanced in order to find a fit among cost, performance, development effort, development time, risk and the application. In the early phase of the process, the basic structure is determined as the architecture of the system, as we will see in Section 2.

As the process continues, the final system is gradually worked out in a manner that can best be described by *successive approximation*. Both in the problem and in the solution domain alternatives are described, compared, and decided upon. This is done in multidisciplinary teams –the times of sequential mechanical, electrical, software design etc. are passé–.

In this paper we treat the vital role of communication. As we will see, communication is essential to create knowledge and consequently reduce complexity. The paper combines findings from literature with those from practice. We will see that there is a conflict between the need and practice in industry and the main stream of research in systems architecting support.

The paper will deal with system architecture and architecting (Section 2). Next, complexity is regarded from an engineering viewpoint (Section 3). Then knowledge and knowledge creation (Section 4), and communication (Section 5) will be treated. It turns out knowledge is created in a social process among individuals. Then, we will address issues that are identified from the daily practice of system architects in several manners (Section 6). Section 7 summarizes the current trend in systems engineering research. Based on the material treated, we will list the main ingredients for successful complex systems architecting in Section 8; these also serve as a basis for the recommendations for further research in the last section.

2. Architecture and Architecting

Any system has an architecture, consciously created, or evolved during many design cycles and updates. IEEE 1471 [1] elaborates on the definition of an architecture as shown in Figure 1, where a clear distinction is made between the

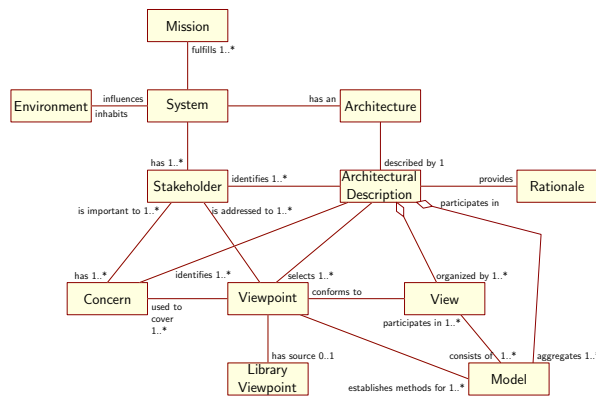


Fig. 1. The function and context of a system architecture, according to IEEE1471²[1].

architecture and architectural description. The term architectural description is further elaborated as “a collection of products to document an architecture”. Note the plural for products. In the accompanying documentation and explanation on the IEEE1471 website [1] an important remark is made: “Lesson: One view isn’t enough, the single hierarchy of components doesn’t describe the real world.” This is also shown by the “aggregates” relation of 1...* between architectural description and model. For instance [2] and [3] illustrate this further.

This shows the architecture of a system is represented in different ways. The products contained in the architectural description are the explicit representations of the architecture that is implemented or being designed in the system, and the relations of the system with the user(s) and context that are part of the Environment in Figure 1, and as elaborated in ISO/IEC/IEEE 42010 which superseded the IEEE1471 standard. As these models of the architectural description may be in different places in documents, diagrams, or even parts and names of the system it can be a challenge to really *know* the architecture. Even more so, one model can express different meanings to different persons (see Section 6.2).

3. Notes about Complexity

The body of knowledge on complexity is large [4–9] to name a few. While size and number of components do play a role, they are not the only determinants of complexity [9,10]. The publication on complexity factors [11] gives a broad overview. This overview also shows that defining complexity is hard, and is sometimes even omitted. For the perspective of this paper we will look at the classification of complexity by Suh [8]. Here complexity is narrowly defined as “the measure of uncertainty in satisfying the FRs³ within their design range”. It is further divided into two major categories [8]:

- *time-independent complexity* divided into two subtypes:

- *time-independent real complexity*: the problem to be solved, or the system to be designed is difficult by nature. An example is the physics involved in a magnetic resonance imaging (MRI) system.
- *time-independent imaginary complexity*: “is defined as uncertainty that is not real uncertainty, but arises because of the designer’s lack of knowledge and understanding of a specific design itself.”
- *time-dependent complexity*, here “future events affect the system in unpredictable ways”. This can be wear in system components, unforeseen system usage, catastrophic events, etc.

Comparing to the findings in [11], time-independent real complexity largely equals objective complexity. Imaginary complexity compares largely to subjective complexity. For the remainder of this paper, we will use the terms real and imaginary complexity as defined by Suh [8].

As we deal in this paper with system architecting, we limit ourselves to the two forms of time-independent complexity. This, however, does not mean that a system designer or system architect should not consider as many as possible improbable events in his design; the time-dependent complexity. This has been illustrated dramatically by the developments in the Japanese Fukushima nuclear power plants. Systems Thinking is a good approach to deal with this type of complexity and provides means to avoid problems [13–15]

From the description of the two subtypes of time-independent complexity, we can conclude that the way to handle complexity is by increasing knowledge. The types and sources of knowledge differ for the two subtypes of time-independent complexity.

In the case of time-independent *real* complexity the technology is difficult by nature. To handle this type of complexity, new knowledge has to be created. In many cases the source of knowledge to handle this type of complexity is fundamental or applied research.

Digesting the definition of imaginary complexity, we can conclude that in this case the difficulty arises from not knowing *enough* about the problem or about the solution; not necessarily because the problem in itself is difficult. This complexity is, one could say, inversely related to the knowledge available to the designers and architects. The remedy here is investigation, knowledge sharing and making the implicit (tacit) knowledge explicit.

The above confirms (and formalises) the observations from architects and researchers, including the author, that multidisciplinary complicates design and architecting. Causes are found in diverse formalisms, different opinions (including opinions of what is difficult), and different approaches (see for instance [16–19]).

Therefore, we will look at both knowledge creation and communication briefly in the next two sections.

4. Knowledge Creation

Nonaka and Takeuchi [20] treat the way organizations create knowledge. They have investigated in particular Japanese organizations. With influences from other cultures, they come to a way of working for organizations in what Peter Drucker

²The standard has been superseded by ISO/IEC/IEEE 42010. Yet for illustration we use IEEE 1471, as it shows Architecture and Architectural Description in one scheme.

³Functional Requirement, according to Suh’s Axiomatic Design theory [12].

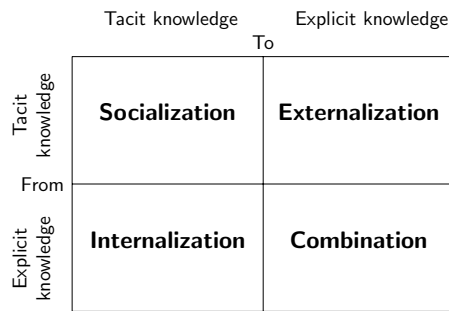


Fig. 2. The knowledge creation processes according to Nonaka and Takeuchi [20]

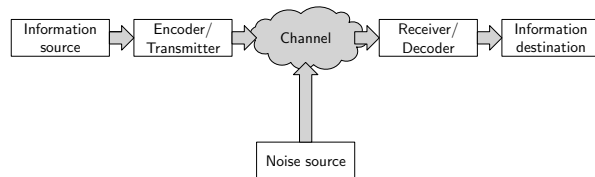
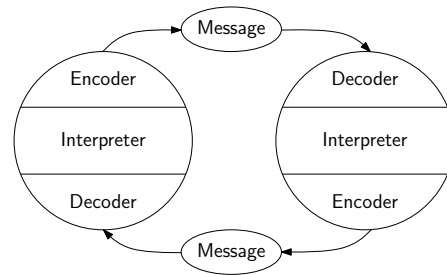


Fig. 3. The Shannon-Weaver model of communication [23].

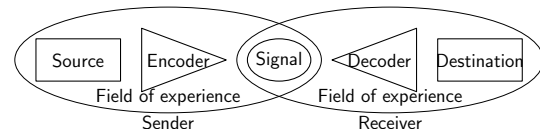
calls the “Knowledge Society”. For their argument, they use the difference between *explicit* knowledge and *tacit* knowledge. Explicit knowledge is that what is handled in documents, presentations, spreadsheets, models, etc. A large part of research in system design and engineering is about manipulating explicit knowledge, be it in the form of SysML models [21], or in requirements management packages, or Design Structure Matrices. This is well dealt with in the expanding body of knowledge on model based design and model based systems engineering (MBSE, see [22] and references contained therein). Tacit knowledge is by definition not in these kinds of models. It is the way a carpenter knows how to use the chisel, it is the way a system designer *projects* the consequences of an architecture decision. Figure 2 shows the knowledge creation process according to Nonaka and Takeuchi. Starting point is either tacit or explicit knowledge. The goal is also either tacit or explicit knowledge. Four processes are shown that link the starting and goal types of knowledge. Nonaka and Takeuchi contend that the key to knowledge creation lies in the mobilization and conversion of tacit knowledge, and that knowledge conversion is a social process *between* individuals; not confined within one individual. This means that for this knowledge creation and conversion to take place, communication between system designers and other people involved in the design process is essential; it enables knowledge to be put into models for further processing in MBSE like approaches. We will therefore look at communication, next.

5. Communication

As we have seen, knowledge creation relies on information and knowledge sharing, hence communication. One of the most well-known models of communication is the Shannon-Weaver model of communication [23], see Figure 3. While designed for investigating technical issues, it has a much wider area of application. It clearly shows the effect of encoding, channel, and decoding. Also noise is clearly shown.



(a) First part of Schramm's model of communication showing feedback, and the encoder and decoder at the sender's and receiver's side.



(b) Second part of Schramm's model of communication showing fields of experience of both the sender and the receiver.

Fig. 4. Schramm's model of communication [24].

However, there is no way of checking the message against its original content, nor ways to correct messages (feedback). This is better shown in the first part of Schramm's model of communication (see Figure 4(a), also known as the Osgood-Schramm model of communication) [24]. The second part of the Schramm model (Figure 4(b)) includes the fact that for successful communication both partners need a common field of experience. Clark and Brennan [25] use the term “common ground” to indicate the shared part of the fields of experience. As the shared part may be small, there may not be an inverse relationship between the decoder and encoder. This may thus lead to communication errors. Fortunately the feedback mechanism shown in Figure 4(a) can be used to adapt the message, and even the encoder and decoder in order to get a closer approximation of the desired inverse relationship; a form of learning feedback.

When applying the above to our field, we can conclude that architecting should take place in a social process among architects and other stakeholders. Communication of, and feedback on, partial or intermediate versions of the architecture is important. To concentrate on the architecting, the architecture and other products should be presented in a common ground format, noise should be reduced, as well as the encoding-decoding kept simple. That is difficult because of the diverse backgrounds of the people involved. In fact, there are three main approaches:

1. *introducing a new formalism,*
2. *using several formalisms* (of which one or more are understandable to each stakeholder), or
3. *using an existing widely understandable formalism.*

6. Architecting in Practice

In this section we will look at examples of architects at work in real life. The information has been obtained in various ways to ensure validity (the triangulation principle in [26]). The goal is to see what works in industry, and what academia can learn.

Table 1. Observations made by architects during interviews (see [17]).

- Architect A:
 - Requirement Engineering Tools are used, though not common. The **advantages** of these tools are traceability and less trouble with loose documents. The **disadvantage** is that the available information is enormous and thus a lot of this information is trivial. All this trivial information has not much use.
- Architect B:
 - Phone: One cannot point out things, a reference is missing
 - E-mail: Here, the reasoning can be made pretty clear, but one is not sure if the receiving party interprets the data as was intended, because there is no direct feedback.
- Architect C:
 - Sometimes misunderstandings arise due to the fact that the document can be interpreted in multiple ways
 - due to the clinical nature of the products, there are rules and regulations for the requirements. This formal notation is however not efficient both in creating and presenting the information later on.
- Architect D:
 - A tool should never take away the responsibility of the systems architect

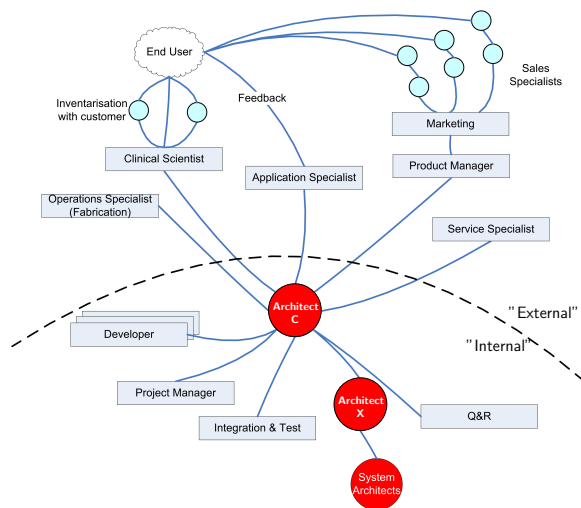


Fig. 5. Stakeholder's context diagram as created with "Architect C".

6.1. Interviews with Architects

A series of interviews with practicing systems architects has been carried out by Haveman [17] within the context of the present research. The architects are very experienced and work at large Dutch and/or multinational companies that develop and produce consumer electronics, medical equipment or security systems. The most important remarks made are summarized in Table 1. Additionally, a context diagram of communication was made with each architect interviewed to visualize all stakeholders the architect communicates with. Only one (Figure 5) is included in this paper due to space reasons. Some terms have been desensitized because of confidentiality. Common in all the diagrams created is that architects have many communication partners with very diverse backgrounds and very diverse interests.

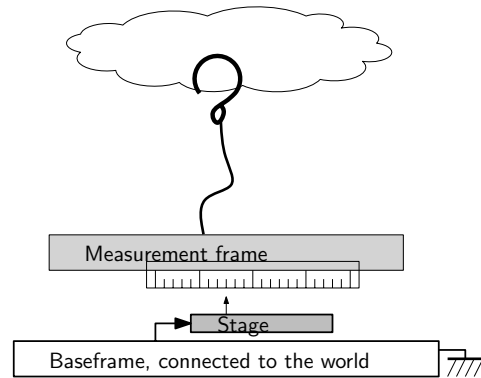


Fig. 6. Pictorial Architecting: an illustration used to clarify the dynamic architecture of a wafer stepper. The hook in the cloud represents the "sky-hook suspension". The small arrow pointing to the ruler represents the position measurement system.

6.2. Group Discussion at CSA2010

In Twente, the Netherlands, in December 2010, people from industry and academia participated in the Second Workshop on Complex Systems Architecting⁴ (CSA2010) to bring together practice and research in systems architecting. Part of the program was group discussions, one of the topics being "Scenarios for architectural communication systems". One discussion was held between seven people, both from academia and industry, both experienced and novice architects, both practitioners and researchers. From the discussion about the current way of working, and present and ideal scenario's, the problematic issue of misinterpretation of an architecting message was identified. The architect sends message X, that is interpreted as Y or Z by other stakeholders, leading to implementation problems.

6.3. Architects at Work

In this subsection, two relatively successful ways of architecture communication are presented. These examples are real-life experiences observed by the author in two different industrial settings. Two more examples are taken from recent literature and a key note presentation.

6.3.1. Pictorial architecting

In one of the first meetings with a system architect, who is one of the most prolific inventors at a former employer of the author, this architect explained a dynamic architecture of the to be designed new wafer scanner. On the white board appeared a very simple figure (like Figure 6). The hook represents a "sky hook suspension". Here, the position of the measurement frame is kept stationary with respect to an inertial reference. Position of the stage is measured relative to this measurement frame, in a non-contact manner. Stage propulsion is done relative to a frame connected to the rigid world.

On other occasions the architect used a similar picture for clarification to other designers and engineers. The clear message did not dive into intricacies of the suspension system, or the challenges for the servo-experts, nor does it complicate the

⁴Proceedings can be obtained from the author.

picture by adding many sensor systems. Using this very simple model (the picture *is* a model), that was repeated for many audiences with very different backgrounds led to the successful development of the architecture of the present line up of wafer steppers [27]. In such an architecture representation it became clear to all system designers and detail designers what the implications are for their design decisions. Therefore deep understanding of this basic model by all involved was essential. Note that the model is not complete, nor is it impressive, nor was it the only model. Yet, it succeeded in telling the core message for the dynamic architecture (using an inertial mass as measurement reference and exerting all forces to the base-frame), and it became one of the common models within the large development team.

6.3.2. *Mathematical modelling*

The second case involves a company developed from the physics department of a technical university. As is known, physicists use mathematics as their common language. As long as the vast majority of the company consisted of people with a physics background, nearly all issues were modelled in mathematics on white boards, or in mathematical modelling computer programs. While a throughput model at the company above used to be a spreadsheet, it was a mathematical model in this company. Even more so, the software code was a direct derivative of the mathematical models created in the analysis and design phase!

It should be noted though, that in later stages of the development process, “experience” and “engineering knowledge” had to be incorporated in order to design working systems quickly enough. Therefore, mechanical and software engineers were hired to introduce that knowledge. This caused some transient behaviour as the mechanical engineers used formulas from the engineering book, where the physicist challenged these and wanted to start from the fundamental physics equations.

6.3.3. *Emergency Response system*

An emergency response system for China is presented in [28]. The way this is done, is by presenting four diagrams. Each of the diagrams can be read by any engineer. Although some formalisms are used from flowcharting, these are not strictly applied. The main goal of the diagrams is to present the essence of the architectures.

6.3.4. *Model Based Systems Architecting*

The Vice President of Schindler Elevators presented the “pragmatic approach to Systems Engineering” at the German Systems Engineering event 2012 [29]. The company implemented a set of tools for Model Based Systems Engineering to be able to calculate any elevator in a top-down manner, and to be able to simulate it. Also exceptions and critical situations can be calculated.

In many of the diagrams the elevator and/or the building is recognisable, while the control diagrams connect to the control engineers “vocabulary”. Further, simulation, overview and details are considered.

6.4. *Summarizing Architecting in Practice*

Common to all the observations above, and in line with the theoretical basis laid in Sections 2–5, is that for successful ar-

chitecting, the stakeholders should be engaged in a fruitful communication process. This should in the first place take place on common ground [25]. In the examples above, these were pictures, mathematics, flowcharts, and the general elevator structure. None of these uses approach (1): the new formalism (see Section 5). Instead, a widely known formalism, or combination of formalisms was used, so approach (3) or (2).

7. **Current Systems Engineering Research**

The main stream of Systems Engineering research is summarized by Seven Grand Challenges of Systems Engineering Research, presented in [30]. These are partially reiterating challenges formulated in 2009. Two of them are technical of nature (Ultra scalable autonomous and heterogeneous systems, respectively) and one relates to the SE business case and competences. The other four promote research in the direction of modelling and simulation. “The desire to accurately model/simulate a complex system and its environment in minute detail through to a higher level of abstraction still remains the ultimate challenge.” These grand challenges are reflected in the many tools presented at conferences and in literature that are centred around one or a limited number of formalisms. The tools promote increasing the level of detail in models, maybe not deliberately. Often correct and complete input is needed before output can be generated, requiring substantial work. The tools and methods are used by architects behind workstations. Interaction between architects is not promoted, and the creation of a hardcopy to use at a meeting is difficult.

8. **Successful Complex Systems Architecting**

Now that the status of architecting is given, the important questions are: Where to go next? How to get there? Is it in line with the identified grand challenges? Are there crucial steps to be made before successful architecting is feasible?

From the examples and the theoretical considerations, the following recommendations for successful complex systems architecting can be described:

- **Multiview:** use various formalisms to connect to the diverse stakeholders involved. Answering to the IEEE 1471 and 42010 multiple view requirements.
- **Simplify as much as possible.** This requires a significant effort from the architect to find the essence. Different representations may be required for different discussions, as the essential components differ.
- **Connect to reality.** The representations should resemble the system under design. An abstract boxology is not enough. At first glance the representation of an MRI system should be different from the one of a wafer stepper.
- **Use the tripod of Functional-Physical-Quantification** in all representations. Functional represents the abstract “what”, physical represents the concrete “how” and “where”, while quantification addresses the “how much”.
- **Use feedback.** An architecture evolves and is developed by involving diverse stakeholders. By presenting intermediate results, tracking the reactions, improving the architecture, the architecture develops in a spiral way. Timeboxing can be useful here.

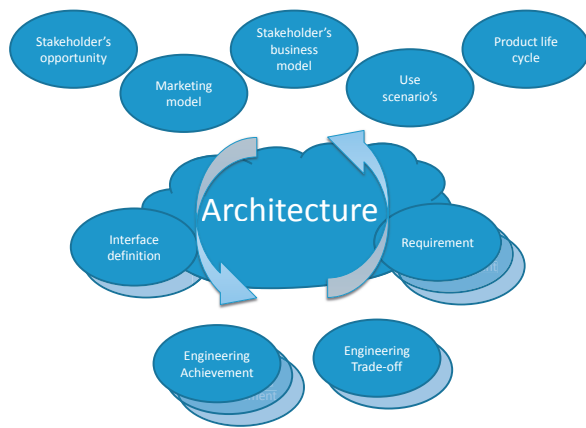


Fig. 7. The central role of the architecture between development, engineering, marketing, and other stakeholders. It has to be used as communication means.

- Use low-order modelling. To support the quantification, numbers are required that base on reality. These can be derived with often easy models. While an error margin has to be taken into account, for the architecture these are accurate enough. When detail design begins, more precise models have to (and will be) developed.
- Communicate! This may well be the most important aspect. The architecture has to be communicated as much as possible, see figure 7. Not only within development, but also to marketing and other departments, just as much as input for the architecture is required from all these stakeholders. In fact, the architecture is the main interface between them: it shows the effect of engineering achievements to marketing, as well as the need of marketing opportunities to development.

This may sound quite like using common sense. Yet, it is not what the grand challenges describe; quite the opposite.

As an alternative, the A3 architecture overviews [16] are a good starting point for the above way of architecting. They are explicitly directed at creating a hardcopy to use at meetings and to accumulate comments to be processed into a new version of the A3. Accepted versions can be distributed and used as posters or placemats [31] for reference in daily work. Additional tools are the CAFCR method (multiview, [32]) and FunKey Architecting (quantification coupled to functional and physical views [33]).

The Architecture Model developed in a joint project between Delft University of Technology and University of Twente [34,35] is a formal way of modelling architecture information. However, as it models connections between information in domain specific formalisms, it provides means to connect to familiar monodisciplinary modellers like mechanical CAD, mathematical modelling, knowledge based engineering, etc. This answers to the multiview way of modelling. While more research is needed, a solid foundation is laid.

This paper and the special session aims at promoting the research effort in the direction of communication and the points raised in the beginning of this section.

9. Further Research

While a lot of research has been put into formalisms for architecting, there are still open issues in light of the above, because the need for documentation is valid. In for instance fields like medical and aerospace, certification requires solid and unambiguous documentation.

One main research track should therefore be directed to solving the opposition between understandability and formality. At this moment with increasing formality, the group of stakeholders that understand the model decreases. While with increasing understandability, the ambiguity increases.

Related to the above is the model flow in architecting projects. From incomplete and uncertain, and even ambiguous, to increasingly more certain and more detailed. The reuse of detailed models is an issues as well. A start of connecting high and low level models has already been made in our group [36].

Also the role of uncertainty in design and architecting requires attention. Blanchard and Fabrycky make mention of the issue [37], but more research is needed, as already initiated by [38].

Acknowledgements

The author thanks Gerrit Muller and Tetsuo Tomiyama for many fruitful discussions on this topic, Steven Haveman for his contributions in the form of interviews with architects.

References

- [1] IEEE, . Ieee 1471 website. Last accessed on 7 november 2013; URL: <http://www.iso-architecture.org/ieee-1471/index.html>.
- [2] Martin, J.N., Ferris, T.L.. On the various conceptualizations of systems and their impact on the practice of systems engineering. In: IS2008. Utrecht, the Netherlands: INCOSE; 2008.
- [3] INCOSE SEH Working Group, . Systems Engineering Handbook. 3.1 ed.; INCOSE; 2008.
- [4] Gell-Mann, M.. What is complexity?. Complexity 1995;1(1):16–19.
- [5] Manson, S.M.. Simplifying complexity: a review of complexity theory. Geoforum 2001;32(3):405–414. URL: <http://www.sciencedirect.com/science/article/B6V68-437XPXC-9/2/28058a2ba63bec48fa116cfa10520f23>.
- [6] McDermid, J.A.. Complexity: concept, causes and control. In: Sixth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2000). IEEE; 2000.
- [7] Nakagawa, T., Yasui, K.. Note on reliability of a system complexity. Math Comput Modell 2003;38(11-13):1365. URL: <http://www.sciencedirect.com/science/article/B6V0V-4BRR7B3-11/2/18feadf8ba6036ce432051e061d49fd1>.
- [8] Suh, N.P. Complexity in engineering. CIRP Ann 2005;2(2005):581–598.
- [9] ElMaraghy, W., ElMaraghy, H., Tomiyama, T., Monostori, L.. Complexity in engineering design and manufacturing. CIRP J Manuf Sci Technol 2012;61(2):793 – 814. URL: <http://www.sciencedirect.com/science/article/pii/S0007850612002004>. doi:<http://dx.doi.org/10.1016/j.cirp.2012.05.001>.
- [10] Tomiyama, T.. Architecture-centric model-based product development. In: Mechatronics 2012. Linz, Austria; 2012.
- [11] Dong-Han, H., Jinkyun, P., Wondea, J.. A framework-based approach to identifying and organizing the complexity factors of human-system interaction. IEEE Syst J 2011;5(2):213–222. URL: <http://dx.doi.org/10.1109/JSYST.2010.2102574>.
- [12] Suh, N.P.. Axiomatic design theory for systems. Res Eng Des 1998;10(4):189–209. URL: <http://dx.doi.org/10.1007/s001639870001>.
- [13] Boardman, J., Sauser, B., John, L., Edson, R.. The conceptagon: A framework for systems thinking and systems practice. In: Systems, Man

- and Cybernetics, 2009. SMC 2009. IEEE International Conference on. 2009, URL: <http://dx.doi.org/10.1109/ICSMC.2009.5346211>.
- [14] Richmond, B.. Systems thinking: Critical thinking skills for the 1990s and beyond. *Sys Dyn Rev* 1993;9(2):113–133. URL: <http://dx.doi.org/10.1002/sdr.4260090203>.
- [15] Bonnema, G.M.. Thinking tracks for integrated systems design. In: 1st Joint International Symposium on System-Integrated Intelligence 2012. Hannover; 2012, p. 32–35. URL: <http://purl.utwente.nl/publications/80880>.
- [16] Borchers Juzgado, P.D.. A3 architecture overviews : a tool for effective communication in product evolution. Ph.d.-thesis; University of Twente; 2010. URL: <http://purl.utwente.nl/publications/75284>.
- [17] Haveman, S.. Project buzz tracker - supporting system architects in the future workspace. Master thesis; University of Twente; 2009. URL: [http://www.futureworkspaces.nl/2009/08/17/project-buzz-tracker/\(summary\)https://doc.novay.nl/dsweb/Get/Document-102557/FWS_Project_Buzz_Tracker_Haveman.pdf\(fullreport\)](http://www.futureworkspaces.nl/2009/08/17/project-buzz-tracker/(summary)https://doc.novay.nl/dsweb/Get/Document-102557/FWS_Project_Buzz_Tracker_Haveman.pdf(fullreport)).
- [18] Borchers, P.D., Bonnema, G.M.. A3 architecture overviews - focusing architectural knowledge to support evolution of complex systems. In: 20th Annual INCOSE International Symposium - IS2010. Chicago; 2010, URL: <http://purl.utwente.nl/publications/77042>.
- [19] Tomiyama, T., DAmelio, V., Urbanic, J., ElMaraghy, W.. Complexity of multi-disciplinary design. *CIRP J Manuf Sci Technol* 2007;56(1):185 – 188. URL: <http://www.sciencedirect.com/science/article/pii/S0007850607000467>. doi:<http://dx.doi.org/10.1016/j.cirp.2007.05.044>.
- [20] Nonaka, I., Takeuchi, H.. *The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press; 1995. URL: <http://www.amazon.com/Knowledge-Creating-Company-Japanese-Companies-Innovation/dp/0195092694>.
- [21] Object Management Group, . The official omg sysml site. Last accessed on 7 november 2013; URL: <http://www.omg-sysml.org>.
- [22] Reichwein, A., Paredis, C.J.. Overview of architecture frameworks and modeling languages for model-based systems engineering. In: *Proc. ASME 2011 International Design Engineering Technical Conf. & Computers and Information in Engineering Conf.* Washington, DC: ASME; 2011..
- [23] Shannon, C., Weaver, W.. *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press; 1949.
- [24] Schramm, W.. *The Process and Effects of Mass Communication*. Urbana, IL: University of Illinois Press; 1954.
- [25] Clark, H.H., Brennan, S.E.. Grounding in communication. In: Resnick, L.B., Levine, J.M., Teasley, J.S.D., editors. *Perspectives on socially shared cognition*. Washington DC: American Psychological Association; 1991, URL: <http://www.psychology.sunysb.edu/sbrennan/papers/clarkbrennan.pdf>.
- [26] Martin, J.N., Davidz, H.L.. Systems engineering case study development. In: *5th Annual Conference on Systems Engineering Research 2007 (CSER2007)*. Hoboken, New Jersey: Stevens Institute of Technology; 2007, URL: <http://sse.stevens.edu/fileadmin/cser/2007/proceedings/9.pdf>.
- [27] Loopstra, E.R., Bonnema, G.M., van der Schoot, H., Veldhuis, P., ter Beek, P.M.. Two-dimensionally balanced positioning device with two object holders, and lithographic device provided with such a positioning device. European Patent Office; EP0890136 DE69717975T; 1999. URL: <http://v3.espacenet.com/textdoc?DB=EP0890136&CLASS=DE69717975T&F=0&QPN=DE69717975T>.
- [28] Shiyong, L.. Employing system of systems engineering in china's emergency management. *IEEE Syst J* 2011;5(2):298–308. URL: <http://dx.doi.org/10.1109/JSYST.2011.2139350>.
- [29] Henneau, P.. A pragmatic approach to se. Key Note presentation at Tag des Systems Engineering, Paderborn, Germany; 2012.
- [30] Kalawsky, R.S.. The next generation of grand challenges for systems engineering research. *Procedia Comput Sci* 2013;16(0):834–843. URL: <http://www.sciencedirect.com/science/article/pii/S1877050913000884>.
- [31] Kooistra, R.L., Bonnema, G.M., Skowronek, J.. A3 architecture overviews for systems-of-systems. In: *Complex Systems Design & Management 2012*. Paris; 2012, URL: <http://purl.utwente.nl/publications/82854>.
- [32] Muller, G.J.. *Cafr: A multi-view method for embedded systems architecting*. Ph.d.-thesis; Delft University of Technology; 2004. URL: <http://www.gaudisite.nl/Thesis.html>.
- [33] Bonnema, G.M.. Insight, innovation, and the big picture in system design. *J Syst Eng* 2011;14(3):223–238. URL: <http://dx.doi.org/10.1002/sys.20174>.
- [34] Woestenenk, K., Cabrera, A.A.A., Bonnema, G.M., Tomiyama, T.. Capturing design process information in complex product development. In: *ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2011)*. Washington DC: ASME; 2011..
- [35] Woestenenk, K., Cabrera, A.A.A., Tragter, H., Tomiyama, T., Bonnema, G.M.. Multi domain design: Integration and reuse. In: *22nd International Conference on Design Theory and Methodology (DTM)*. Montreal, Quebec, Canada: ASME; 2010..
- [36] Haveman, S.P., Bonnema, G.M.. Requirements for high level models supporting design space exploration in model-based systems engineering. *Procedia Comput Sci* 2013;16(0):293–302. URL: <http://www.sciencedirect.com/science/article/pii/S187705091300032X>.
- [37] Blanchard, B.S., Fabrycky, W.J.. *Systems Engineering and Analysis*. Fifth ed.; Upper Saddle River, New Jersey: Prentice Hall; 2011. URL: <http://www.pearsoned.co.uk/bookshop/detail.asp?item=100000000374477>.
- [38] Rajabalinejad, M., Spitas, C.. Incorporating uncertainty into the design management process. *J Des Man* 2011;6(1):52–67. URL: <http://dx.doi.org/10.1111/j.1948-7177.2011.00022.x>.